Clasificación de datos usando el método k-nn

Jorge Enrique Rodríguez Rodríguez[1]

Edwar Alonso Rojas Blanco[2]

Roger Orlando Franco Camacho[3]

Resumen

En este artículo se muestra cómo usar el método k-nn (k-nearest neighbours o en español k-vecinos más cercanos) para clasificación de datos mediante la descripción del proceso de desarrollo del programa k-nn v1.9 desarrollado en la Facultad Tecnológica de la Universidad Distrital Francisco José de Caldas. Se parte de las bases teóricas de k-nn, luego se describe el algoritmo que se implementó durante el desarrollo del software y cómo se sortearon los diferentes problemas durante su implementación, y finalmente se analizan los resultados de las pruebas realizadas con la aplicación k-nn v1.9, de donde se obtienen algunas conclusiones.

Palabras clave

Vecindad, heurística de consistencia, distancia euclidiana, atributo, clase, casos o instancias, datos de entrenamiento, radio, efectividad, matriz de confusión, eficiencia.

1 Ingeniero de Sistemas, especialista en Telemática, especialista en Ingeniería de Software, magíster en Ingeniería de Sistemas, docente investigador de la Universidad Distrital Francisco José de Caldas, director del Grupo de Investigación en Inteligencia Artificial, jrodri@udistrital.edu.co.

2 Tecnólogo en Sistematización de Datos de la Universidad Distrital Francisco José de Caldas, edwredx@yahoo.es.

3 Tecnólogo en Sistematización de Datos de la Universidad Distrital Francisco José de Caldas, rofstai@yahoo.es.

Abstract

This paper shows how to use the k-nn method (k- nearest neighbour) for data classification, by means of the description of the process of development of the program k-nn v1.9, developed in the Technological Faculty of the Francisco José de Caldas Distrital University. The theoretical bases of k-nn are the starting point, then the algorithm used in the software and the way the differenrent problems were solved during its implementation are described, and, finally, the results of the tests carried out with the application k-nn v1.9 are analyzed, upon where conclusions are drawn.

Keywords

Vicinity, heuristic of consistency, Euclidean distances, attribute, class, cases or instances, data of training, radio, effectiveness, confusion matrix, efficiency.

Introducción

El método k-nn pertenece al grupo de métodos para tareas de clasificación de datos que se pueden encontrar dentro de minería de datos. Más específicamente, k-nn es un método de vecindad basado en casos o instancias. Para poder entender cómo clasificar datos usando k-nn es importante abordar temas como el aprendizaje basado en casos (haciendo énfasis en el concepto de heurística de consistencia), los métodos basados en vecindad y algunos tipos de medición de distancia.

Con las bases teóricas mencionadas anteriormente fue posible desarrollar el software k-nn v1.9, que es un prototipo de software que clasifica datos y cuya base fundamental es precisamente el algoritmo k-nn.

Para dar cuenta de la efectividad del método, durante el artículo se describirán pruebas y se analizarán resultados obtenidos después de clasificar datos usando el programa k-nn v1.9.

En cuanto a la eficiencia del método, que para este caso se representa en gasto o requerimiento de máquina, se analizó el algoritmo utilizado en k-nn v1.9 a través de un análisis y cálculo de complejidad algorítmica.

1. Aprendizaje basado en casos

El aprendizaje basado en casos o instancias consiste en extraer información de un conjunto de datos (también llamados casos o instancias) conocidos y usarla para clasificar nuevos datos o para agrupar datos existentes. Un concepto muy importante dentro del aprendizaje basado en casos es el de heurística de consistencia (la heurística de consistencia es la base del aprendizaje basado en casos) que puede definirse con la siguiente descripción:

Siempre que se quiera adivinar una propiedad de algo, sin que se disponga de otra cosa más que un conjunto de casos de referencia, halle el caso más parecido, con respecto a

propiedades conocidas y del cual se conoce la propiedad buscada. Deduzca que la propiedad desconocida es la misma que la propiedad conocida. (Winston, 1994, p. 428).

Para una mejor comprensión véase el siguiente ejemplo, que tiene dos datos de entrenamiento y uno para clasificar. El primer caso es una canción, que tiene un compás de ¾, se interpreta con instrumentos de cuerda, no posee percusión, es oriunda de la zona andina y su ritmo es pasillo. El segundo caso de entrenamiento es otra canción en compás partido, interpretada con instrumentos de viento, incluye percusión, es oriunda de la costa atlántica colombiana y es de ritmo cumbia. Usando el criterio de heurística de consistencia se desea adivinar el ritmo de una canción con las siguientes características: compás partido, se interpreta con instrumentos de cuerda, incluye percusión y es originaria de la costa atlántica. La respuesta sería "cumbia", ya que de los dos datos de entrenamiento conocidos el dato en cuestión comparte más características con el segundo, por lo tanto éste y el dato a clasificar pertenecen a la misma clase (cumbia).

Del ejemplo anterior se pueden extraer los conceptos atributo (que serían compás, instrumentos y origen) y clase (que es también un atributo, en el ejemplo anterior el atributo clase sería ritmo). Los atributos son las diferentes características que determinan un dato, es decir, lo particularizan o diferencian de otros. La clase es un atributo que sobresale de los demás y es la base de la que se parte para poder clasificar y agrupar instancias, ésta es la naturaleza del dato.

2. Razonamiento basado en casos

Una de las grandes ventajas de la inteligencia humana sobre la inteligencia artificial es el sentido común. La potencia del razonamiento a través del sentido común es muy grande para la resolución de problemas. Para que una persona razone a partir del sentido común es necesario que durante el transcurso de su vida haya adquirido conocimiento (que puede estar en la memoria representado como experiencia) y sepa relacionar este conocimiento con las situaciones que afronte en su presente.

Normalmente un programa de computadora que analice, por ejemplo, la trayectoria de un proyectil lanzado desde un cañón, usaría las clásicas fórmulas de movimiento uniforme acelerado para calcular las diferentes dimensiones del movimiento, pero un programa de inteligencia artificial que se basara en algo parecido al sentido común humano seguramente determinaría el comportamiento del lanzamiento con base en las características del proyectil y en experiencias de lanzamientos anteriores, posiblemente como lo harían la mayoría de personas, dado que no conocen las fórmulas de movimiento uniforme acelerado y simplemente recurrirían a su conocimiento y lo relacionarían con la situación actual para dar solución al problema.

El razonamiento basado en casos o CBR (Case Based Reasoning) trata de imitar el sentido común humano basándose en experiencias pasadas o casos y de solucionar problemas presentes a través de analogías con problemas pasados. En el libro *Inteligencia artificial,* de Elaine Rich, se plantean cuatro preguntas que deben ser respondidas por un sistema CBR para que actúe correctamente (Rich, 1994, p. 602):

1. ¿Cómo están organizados en la memoria los diferentes casos?
2. ¿Cómo se pueden recobrar los casos relevantes almacenados en la memoria?
3. ¿Cómo se pueden adaptar casos anteriores a los nuevos problemas?

4. ¿Cómo se almacenan ordinalmente los casos?

Para la solución de problemas con CBR no basta únicamente con tener casos en la memoria, sino que además deben estar muy bien organizados para poder acceder a ellos y relacionarlos con un problema actual. Pero además de la organización es importante tener en cuenta los casos más relevantes y parecidos con el problema, ya que éstos brindarían al sistema valiosa información heurística que puede usarse para determinar una solución, como lo haría por ejemplo un médico, quien relacionaría cada nuevo caso con casos anteriores con características similares, y en especial aquel caso más parecido, para establecer un posible diagnóstico.

El CBR es un modelo que podría llevar a las máquinas a aproximarse al sentido común humano, permitiéndoles así hallar soluciones a problemas del mundo real, que es análogo y no tan lineal como se le modela matemáticamente.

3. Métodos basados en vecindad

Los métodos basados en vecindad son fundamentalmente dependientes de la distancia y en consecuencia poseen características propias de ésta como la cercanía, la lejanía y la magnitud de longitud, entre otras.

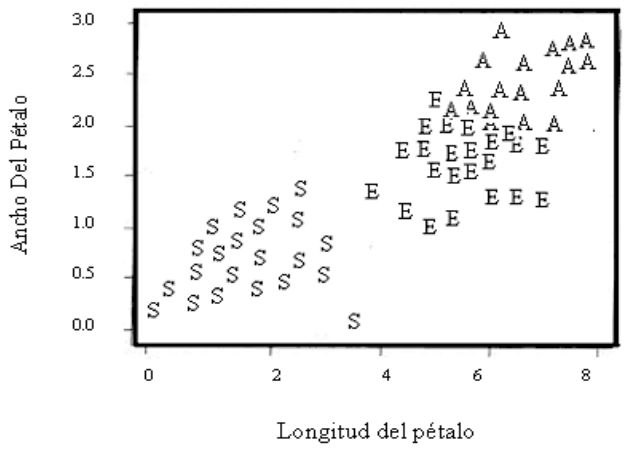
Los métodos basados en vecindad, además de servir para tareas de clasificación, también se usan para agrupación de datos, sin embargo en este artículo únicamente se tratará el tema de la clasificación. Existen dos grupos de métodos de vecindad, según la forma en que se realiza el aprendizaje. El grupo de los métodos retardados (o *lazy*) y los no retardados (o *eager*). En los métodos retardados como k-nn, cada vez que se va a clasificar un dato, en la fase de entrenamiento, se elabora un modelo específico para cada nuevo dato, y una vez que éste se clasifica sirve como un nuevo caso de entrenamiento para clasificar una nueva instancia. En los métodos no retardados se generaliza un solo modelo (también a partir de casos conocidos) para todos los nuevos datos que se desean clasificar, y éstos únicamente son tomados en cuenta como datos de entrenamiento cuando se vuelve a construir un nuevo modelo general. Para ver más claramente la diferencia entre los métodos retardados y los no retardados, veamos un ejemplo de "discriminantes lineales" (método no retardado), y "k-vecinos" (método retardado).

Discriminantes lineales

En el siguiente ejemplo (Hernández Orallo, Ramírez Quintano, y Ferri Ramírez, 2004, p.426) se muestra la forma de clasificación usando discriminantes lineales de Fisher. Se tiene un grupo de datos que se representan como puntos en un plano (como se muestra en el gráfico 1), los cuales hacen referencia a tres variedades de lirios: setosa (que se representa con la letra S), versicolor (que se representa con E) y virginica (representada con A). En el gráfico 1 se muestran dos atributos (longitud y ancho de pétalo) y la clase (tipo de lirio). Los atributos "longitud de pétalo" y "ancho de pétalo" (medidos en cms) constituyen los ejes x y y del plano, respectivamente.

Para empezar a construir un modelo general para clasificación de datos a partir de los casos iniciales que se tienen, se calculan puntos medios (centroides) en cada aglomeración de datos de cada clase (el que los datos parezcan estar agrupados en forma natural demuestra coherencia de los datos de entrenamiento, para este caso). Posteriormente se calcula la distancia en línea recta entre cada

Gráfico 1. Conjunto de datos para clasificar.



centroide y luego se divide el espacio trazando rectas perpendiculares a las líneas mencionadas anteriormente. Estas rectas perpendiculares deben cortar justo en el centro las líneas que unen los centroides. El modelo resultante es el mostrado en el gráfico 2.

Como se observa, el modelo general que se ha construido a partir de los datos de entrenamiento consiste en una división del espacio en regiones claramente delimitadas, las cuales servirán de criterio para la clasificación de nuevos datos. Por ejemplo, consideremos que se desea clasificar un lirio cuya longitud y ancho de pétalo es de 2 cms. De acuerdo al modelo mostrado en el gráfico 2, es claro que el nuevo dato sería clasificado como "S" pues el punto coordenado conformado por los valores dados a los atributos se ubica dentro de la zona en la que hay una presencia mayoritaria de puntos S.

Como vemos, para clasificar nuevos puntos no es estrictamente necesario volver a trazar nuevos modelos, pues éstos se pueden ir clasificando de acuerdo al modelo establecido inicialmente.

K-vecinos

Consideremos la misma situación de los lirios descrita en el ejemplo anterior. Para clasificar un nuevo dato con el método k-nn se haría lo siguiente: primero, se ubica el dato a clasificar en el plano, supongamos que sus coordenadas son 7 y 2 de longitud y anchura respectivamente (ver gráfico 3). Segundo, se determina un "radio de vecindad". El valor de este radio puede ser asignado a partir de alguna heurística conocida. Tercero, se traza una circunferencia cuyo centro es el dato a clasificar; la circunferencia deberá encerrar uno o varios casos de entrenamiento cercanos a la incógnita, si ninguno queda encerrado significa que la heurística usada para seleccionar el valor del radio no sirve y debe ser cambiada. Cuarto, se determina el valor de k (que también puede estar basado en una heurística), es decir, se establece si se va a comparar con el primer vecino más cercano, 1-nn, o con los dos vecinos más cercanos, 2-nn; en fin, se le da un valor a k. Quinto, asignar la clase al nuevo elemento de acuerdo al valor de k y al número de datos encerrados en la circunferencia.

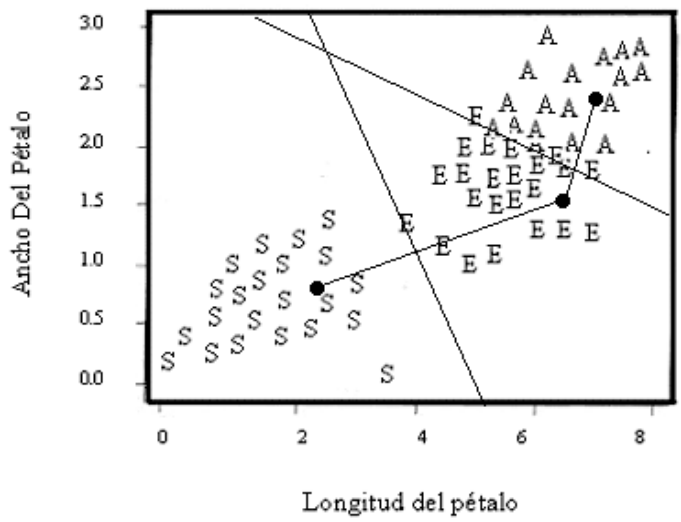


Gráfico 2. Clasificación mediante discriminantes de Fisher.

En el gráfico 3 se han tomado dos radios diferentes con respecto al dato a clasificar. En la circunferencia con menor diámetro hay un individuo de la clase A y un individuo de la clase E. En este caso la clase que tomaría el nuevo dato sería la de la primera instancia más cercana del que supongamos es el A. Si el nuevo caso se quisiera clasificar con respecto a la circunferencia cuyo diámetro es mayor, el dato se clasificaría como A nuevamente, pues son mayoría dentro del vecindario seleccionado, ya que hay tres casos de la clase E y cinco de la clase A.

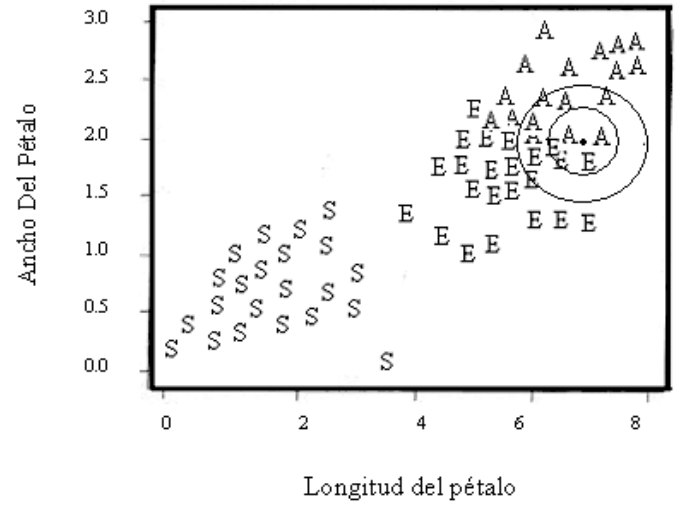


Gráfico 3. Clasificación mediante vecino más cercano.

4. Métricas para medir distancia

La distancia es el criterio de comparación principal usado en los métodos basados en vecindad, por eso es conveniente mencionar algunas de las diferentes formas usadas para su medición. A continuación sólo se mostrarán los modelos matemáticos generales de cada métrica, sin detalles, ya que no es un propósito de este artículo ahondar sobre el tema: simplemente se desea recordar que además de la distancia clásica euclidiana existen métricas alternativas.

$$d(x,y)=\sqrt{\sum_{i=1}^{n}(x_i-y_i)^2}$$

Ecuación 1.Distancia euclidiana.

$$d(x,y)=\sum_{i=1}^{n}|x_i-y_i|$$

Ecuación 2. Distancia de Manhattan.

$$d(x,y)=\max_{i=1\ldots n}|x_i-y_i|$$

Ecuación 3. Distancia de Chebychev.

$$d(x,y)=ar\cos\left(\frac{x^T y}{\|x\|\cdot\|y\|}\right)$$

Ecuación 4. Distancia del coseno.

$$d(x,y)=\sqrt{(x-y)^T S^{-1}(x-y)}$$

Ecuación 5. Distancia de Mahalanobis.

$$d(x,y)=\omega\sum_{i=1}^{n}\delta(x_i,y_i)$$

Ecuación 6. Distancia usando la función delta.

$$d(x,y)=\frac{|x\cup y|-|x\cap y|}{|x\cup y|}$$

Ecuación 7. Distancia entre dos conjuntos.

La distancia euclidiana fue la seleccionada para el desarrollo de k-nn v1.9. La distancia usando la función delta sirve para hallar la distancia entre atributos nominales muy comunes en minería de datos.

5. Clasificación de datos con el método k-vecinos

El método de los k-vecinos o k-nn es un método retardado y supervisado (pues su fase de entrenamiento se hace en un tiempo diferente al de la fase de prueba) cuyo argumento principal es la distancia entre instancias. Tal como se observó en el ejemplo de k-nn presentado anteriormente, el método básicamente consiste en comparar la nueva instancia a clasificar con los datos k más cercanos conocidos, y dependiendo del parecido entre los atributos el nuevo caso se ubicará en la clase que más se acerque al valor de sus propios atributos (cumpliendo así lo planteado por el concepto de heurística de consistencia).

La principal dificultad de este método consiste en determinar el valor de k, ya que si toma un valor grande se corre el riesgo de hacer la clasificación de acuerdo a la mayoría (y no al parecido), y si el valor es pequeño puede haber imprecisión en la clasificación a causa de los pocos datos seleccionados como instancias de comparación.

Para enfrentar este problema se plantearon diferentes variaciones del método: en cuanto a la forma de determinar el valor de k, por ejemplo 1-nn, que no es otra cosa más que usar como instancia de comparación al primer vecino más cercano encontrado.

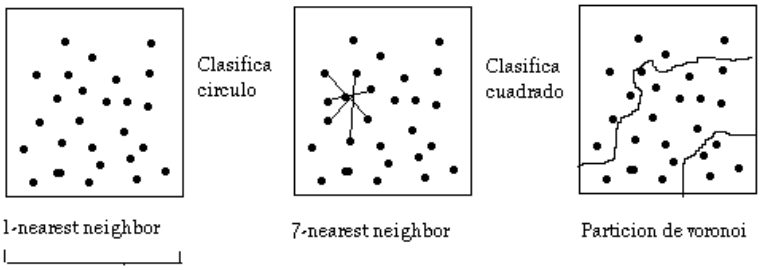


Gráfico 4. Diferencia en el valor de k de los vecinos más próximos y partición realizada.

También el valor de k puede hallarse tomando un radio de comparación o mediante el uso de diagramas de Voronoi.

Una característica importante e interesante de k-nn es que el método puede cambiar radicalmente sus resultados de clasificación sin modificar su estructura, solamente cambiando la métrica utilizada para hallar la distancia. Por lo tanto, los resultados pueden variar tantas veces como métodos de hallar distancia entre puntos haya. La métrica debe seleccionarse de acuerdo al problema que se desee solucionar. La gran ventaja de poder variar métricas es que para obtener diferentes resultados el algoritmo general del método no cambia, únicamente el procedimiento de medida de distancias.

6. Desarrollo de la herramienta k-nn v1.9[1]

Una vez claros los diferentes conceptos del método de clasificación k-nn, ¿cómo empezar a desarrollar un software que clasifique datos usando el método?

Lo primero es seleccionar el tipo de distancia con la que se va a trabajar. Para el caso de k-nn v1.9 se seleccionó la distancia euclidiana.

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^{n} a_1(x_i) - a_r(x_j)^2}$$

Ecuación 8. Distancia euclidiana.

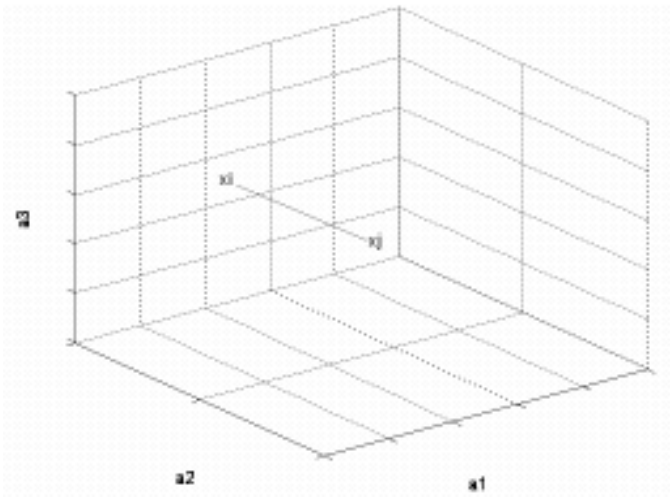
El gráfico 5 representa dos puntos o datos en un plano en R3. La figura será interpretada de la siguiente manera:

Los ejes o dimensiones (a1, a2, a3), representan atributos de los datos. La distancia euclidiana entre los puntos xi y xj (que son dos casos particulares de un conjunto de datos) es igual a la longitud de la recta que los separa. Cuando el número de atributos es mayor que 3 no es posible su representación en un plano, pero la ecuación 8 se sigue aplicando, ya que de todos modos se siguen relacionando los datos a través de sus atributos. En este caso debemos olvidarnos de una interpretación "espacial", como la mostrada en el gráfico 5, y simplemente asumir el resultado de la ecuación 8 como una relación entre atributos de dos instancias particulares.

Una desventaja de únicamente usar la distancia euclidiana es que limita a que todos

1 k-nn v1.9 es un software en lenguaje Java que clasifica datos con base en el método k-nn y que fue desarrollado en el Grupo de Investigación en Inteligencia Artificial de la Universidad Distrital Francisco José de Caldas.

Gráfico 5. Representación de la distancia euclidiana en un plano en R3.



los atributos deben ser de tipo numérico. Adicional a esto, otra delimitación de la aplicación k-nn v1.9 es que tanto los datos de entrenamiento como los casos a clasificar deben estar exentos de ruido, es decir que deben conocerse valores para todos sus atributos: si en una instancia nueva el valor del atributo clase no se conoce, todos los demás sí.

Una vez se interpretó la fórmula de la distancia euclidiana y se adaptó a los términos de k-nn, se estableció un formato de archivo plano (.knn) de entrada de datos (muy parecido al formato .arff usado en la herramienta Weka) para la aplicación, se definieron las herramientas de análisis que se construirían y se desarrolló la función principal del algoritmo. La estructura de dicha función se muestra en el diagrama 1. Las instrucciones (1), (2) y (3) son para ordenar los datos del archivo plano (en el que va toda la información referente a los atributos, además de todos los datos de entrenamiento y los datos a clasificar) en arreglos (matrices y vectores) con el fin de manipular y procesar más fácilmente la información contenida en el archivo. En la instrucción (4) se normalizan todos los datos de entrenamiento, es decir que los valores de los atributos se dejan entre 0 y 1 con el fin de evitar que los resultados de las distancias se queden en valores medios y deterioren los resultados de la clasificación. Para normalizar se usa la siguiente formula:

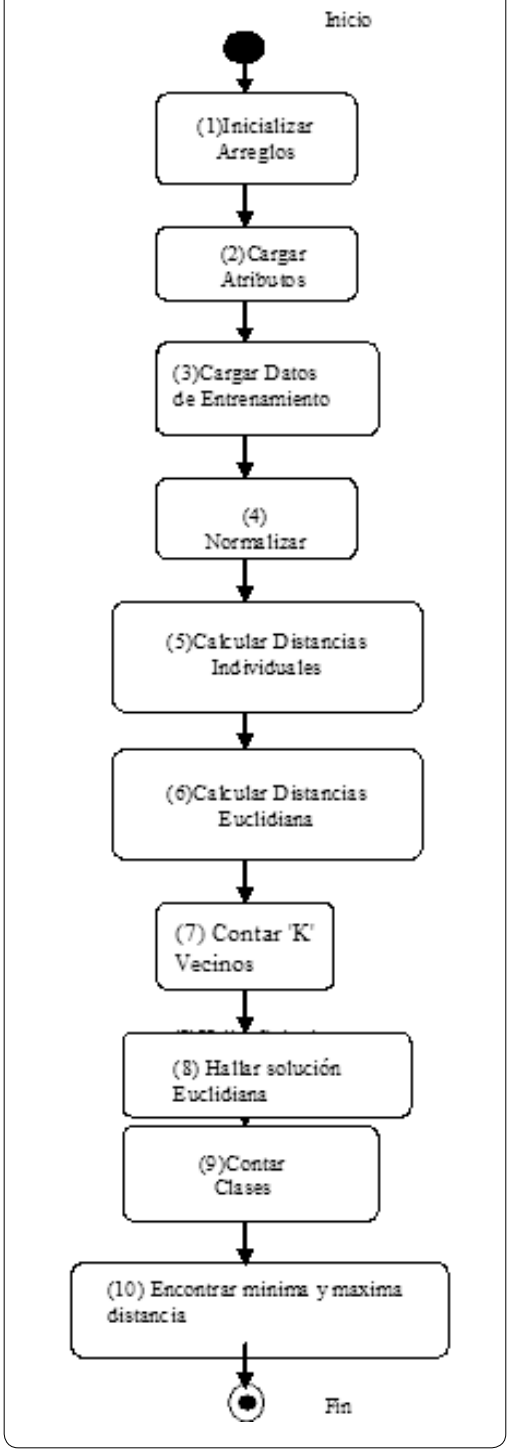
$$V = \frac{(v' - \min A)}{(\max A - \min A)}(nuevoMaxA - nuevoMinA) + nuevoMinA$$

Ecuación 9. Normalización de datos.

Donde V es el nuevo valor a hallar para cierto dato perteneciente al atributo A, v' es el valor actual del dato antes mencionado, minA y maxA son los valores mínimo y máximo de datos que pertenecen a A respectivamente y nuevoMaxA y nuevoMinA son los nuevos valores máximo y mínimo que podrá tomar V. Para este caso esos valores son 0 y 1.

En la instrucción (5) se calcula la diferencia entre los datos de entrenamiento y el dato a clasificar por cada atributo. Por ejemplo, se tienen casos con dos atributos y la clase, y se tienen dos datos de este tipo: A = (10, 20, clase1) –dato de entrenamiento– y B = (5 ,7, ?) –dato a clasificar–, entonces lo que haría el procedimiento (5) sería hallar las siguientes diferencias 10-5 = 5 y 20-7 = 13.

Diagrama 1. Estructura de la función principal de k-nn v 1.9.



La instrucción (6) halla la distancia euclidiana entre el nuevo dato a clasificar y cada dato de entrenamiento. Retomando el ejemplo anterior este procedimiento hallaría el siguiente valor: $d(A,B) = \sqrt{5^2 + 13^2} = 13.92$.

El procedimiento (7) halla el valor de k de acuerdo con el radio seleccionado (el valor del radio es un parámetro del sistema que se pide al usuario antes de comenzar con la clasificación). En la instrucción (8) se halla el caso más cercano a la incógnita y la clase con mayor número de datos dentro del radio seleccionado, de acuerdo con esto se retornan al sistema las soluciones 1-nn y k-nn encontradas. Dentro de las instrucciones (9) y (10) se encuentran funciones necesarias para los resultados de la clasificación y otros análisis que se pueden hacer con k-nn v1.9.

Si se quisiera cambiar la métrica para calcular la distancia (como se sugiere en la sección anterior) lo que se le cambiaría al algoritmo serían las instrucciones (5) y (6), sustituyéndolas por los cálculos correspondientes a la nueva métrica seleccionada.

k-nn v1.9 ofrece las siguientes herramientas:

Clasificación única: Es la clasificación de un solo dato a partir de varios datos de entrenamiento. Esta opción es la principal tarea del programa.

Análisis gráfico en el plano de dos dimensiones: Esta opción funciona solamente después de haber usado la anterior herramienta y consiste en representar como un punto en el plano cada dato de entrenamiento. El origen de este plano es el dato a clasificar. La solución que se dio para representar datos con más de dos atributos en el plano fue asignar la mitad de atributos a un eje y la otra mitad al otro eje. Esto hace posible una aproximación gráfica de la situación de los datos de entrenamiento con respecto al caso a clasificar.

Resultados en gráfico de barras: Una vez se aplica el procedimiento de clasificación de un dato, se pueden ver los resultados obtenidos y las probabilidades estadísticas de que el caso no clasificado pertenezca a determinada clase.

Clasificación múltiple: Esta opción permite aplicar consecutivamente el procedimiento k-nn a varios datos a clasificar. En este caso cada nuevo dato clasificado se toma como dato de entrenamiento para el caso que sigue por clasificar.

Matriz de confusión: Para determinar la efectividad de la clasificación y la calidad de los datos de entrenamiento que se tienen, k-nn v1.9 permite generar matrices de confusión a partir de datos que previamente están clasificados. Lo que hace el programa es tomar un porcentaje de datos para entrenamiento y otro para prueba, luego compara los resultados obtenidos al aplicar el procedimiento k-nn con los datos originales y de acuerdo a los resultados de la comparación se determina un porcentaje de efectividad y otro de error.

7. Pruebas realizadas con el método k-nn

Usando la herramienta de generar matrices de confusión de la aplicación k-nn v1.9, se realizó la prueba descrita a continuación: Se seleccionó un archivo con formato adecuado para el programa con datos de tres clases de lirios: setosa (que se representa con la letra S), versicolor (se representa con E) y virginica (representada con A). A diferencia del ejemplo mostrado en la sección 3 de este artículo (métodos basados en vecindad), aquí se contó con cuatro atributos que son: ancho de pétalo, largo de pétalo, ancho de tallo y largo de tallo. El archivo posee 149 datos en total, de los cuales para una primera prueba se tomó el 70% (es decir 104) para entrenamiento y el 30% (es decir 45) restante como datos de prueba, y se le dio la instrucción al programa de que tuviera en cuenta la solución 1-nn. El resultado obtenido se muestra a continuación.

Tabla 1. Resultados de prueba, con solución 1nn.

	E	S	A	Total	Efectividad
E	5	0	0	5	100%
S	0	2	0	2	100%
A	7	0	31	38	81.57 %
Total	12	2	31	45	93.85 %

De los cinco datos para clasificar de clase E, acertó en cinco, es decir fue un 100% efectivo para clasificar los datos de esta clase. De los 2 datos de clase S que debía clasificar acertó en 2, también con un 100% de efectividad. De los 38 datos que debía clasificar como A, clasificó 7 como E y acertó en 31, por lo que fue efectivo en un 81.57% de los datos y falló en un 18.42%. En general, de todos los 45 datos de prueba clasificó 12 como de clase E, 2 de clase S y 31 de clase A. El promedio de efectividad en general fue de 93.85%. Los desaciertos constituyeron un 6.14% de los datos de prueba.

Veamos a continuación los resultados obtenidos en una prueba usando el mismo grupo de datos, pero teniendo en cuenta la solución k-nn (en donde k depende del número del valor del radio que se asigne) y luego se analizarán los resultados de esta prueba y de la anterior. Al igual que en la prueba anterior de los 149 datos, en total se tomó (aleatoriamente) el 70% de los datos para entrenamiento y el 30% de los datos para clasificar y comparar. Teniendo en cuenta la máxima y mínima distancia encontrada, se designó un valor al radio de 0.3. Los resultados obtenidos se muestran en la tabla 2.

De los 5 datos para clasificar de clase E acertó en cinco, es decir fue un 100% efectivo. De los 2 datos de clase S que debía clasificar acertó en 2, también 100% de efectividad. De los 38 datos que debía clasificar como A, clasificó 16 como E y acertó en 22, por lo que fue efectivo en un 57.89% de los datos y falló clasificando un 42.10%. En general, de todos los 45 datos de prueba clasificó 21 como de clase E, 2 de clase S y 22 de clase A. El promedio de efectividad en general fue del 85.96% y los desaciertos constituyeron en promedio un 14.03% de los datos de prueba. Como se aprecia, los promedios de efectividad de las pruebas anteriores son buenos, sin embargo es preciso resaltar que son sólo promedios y como tal lo generalizan todo. Si observamos por ejemplo más concientemente la tabla 2 vemos que para dos de las tres clases (E y S) la clasificación fue excelente pues tuvo una efectividad del 100%, mientras que la efectividad al clasificar datos de la clase A fue del 57.89%, es decir muy regular. El tener un promedio general de 85.96% de efectividad no es muy diciente pues no nos hace caer en la cuenta de lo bien que funcionó el método para un tipo de datos y lo pésimo que fue para el otro tipo.

Tabla 2. Resultados de prueba con solución k-nn.

	E	S	A	Total	Efectividad
E	5	0	0	5	100.00%
S	0	2	0	2	100.00%
A	16	0	22	38	57.89%
Total	21	2	22	45	85.96%

Al comparar las dos pruebas realizadas nótese que el algoritmo fue más efectivo cuando se seleccionó la solución 1-nn que cuando se eligió la solución k-nn. Sin embargo, no es posible asegurar que el primer tipo de solución es mejor que el segundo, ya que en los resultados pudieron incidir diferentes circunstancias: por ejemplo, que la calidad de los datos de prueba no fuera la mejor o que éstos fueran muy dispersos, o simplemente que no se seleccionó un adecuado valor de radio (o sea un adecuado valor para k), entre muchas otras. Con base en las pruebas anteriores se puede concluir que el algoritmo es más efectivo cuando el valor del radio es pequeño, lo suficiente para determinar un valor de k mayor que uno.

En cuanto a la eficiencia del algoritmo, es decir el rendimiento de la máquina en que se ejecuta, se le realizó un análisis de complejidad algorítmica que llevó a los resultados descritos en la siguiente sección.

8. Análisis de complejidad algorítmica de k-nn

Luego de realizar el análisis de complejidad algorítmica a los procedimientos (5), (6), (7) y (8) (que consideramos como los procedimientos clave y que generan un mayor costo computacional) de la función principal, mostrada en la sección 5 (gráfico 4) de este artículo, se llegó a lo siguiente: las instrucciones (5), (6) y (7) presentaron costos de la forma $T(n) = an^2 + bn + c$, en donde a, b y c dependen de los costos de hardware correspondientes a cada instrucción del procedimiento y n es el número de datos de entrenamiento, por lo tanto el orden de complejidad que se halló en estos 3 procedimientos es de $\Theta(n^2)$, en el peor de los casos.

Por su parte el procedimiento (8) presentó un menor costo computacional que las anteriores instrucciones. La forma de este costo fue $T(n) = an + bm + c$ en donde n es el número de datos de entrenamiento, m el número de clases vecinas, y a, b y c corresponden a los costos de cada instrucción del

Tabla 3. Efectividad de cada uno de los métodos.

	EFECTIVIDAD					
	Naive Bayes	**LWL**	**LMT**	**ZeroR**	**1-nn**	**k-nn**
Contact Lenses	38%	38%	38%	38%	85%	85%
Glass	44%	42%	70%	30%	65%	50%
Kc	69%	74%	70%	64%	82%	82%
Monk	79%	77%	79%	40%	70%	78%
tlak_vse	83%	83%	85%	67%	64%	80%
Kc2	68%	73%	69%	63%	68%	70%
Pc	89%	93%	91%	93%	92%	90%
Vehicles	41%	42%	35%	21%	40%	38%
Donors-train	91%	75%	92%	48%	77%	71%
Text-story	51%	54%	54%	29%	48%	42%
PROMEDIO	65%	65%	68%	49%	69%	69%

procedimiento. De acuerdo con la forma presentada este procedimiento, en el peor de los casos, es de orden $\Theta(n)$, porque siempre n>m. Como vemos el costo computacional depende en gran medida del número de datos de entrenamiento que se tengan: a mayor número de datos de entrenamiento el algoritmo generará mayor costo computacional.

En la tabla 3 se muestra la efectividad para diez conjuntos de datos usando seis métodos de clasificación. Los ejemplos fueron tomados de la web y se probaron con la herramienta Weka (con el fin de probar los métodos Naive Bayes, LWL, LMT y ZeroR) y k-nn v1.9 (para probar con 1-nn y k-nn).

En la tabla se muestran algunas características de los conjuntos de datos utilizados para medir la efectividad del algoritmo implementado.

Tabla 4. Características de los datos tomados como ejemplos de comparación. E: datos para entrenamiento, P: datos de prueba.

	Cantidad de atributos	Cantidad de clases	Total de datos	Datos de entrenamiento 70%	Datos de prueba 30%
Contact Lenses	5	3	23	16	7
Glass	10	6	213	149	64
Kc	22	2	429	300	129
Monk	7	2	123	86	37
tlak_vse	10	2	456	319	137
Kc2	22	2	129	90	39
Pc	22	2	1109	776	333
Vehicles	19	4	846	592	254
Donors-train	53	2	801	560	241
Text-story	150	4	103	72	31

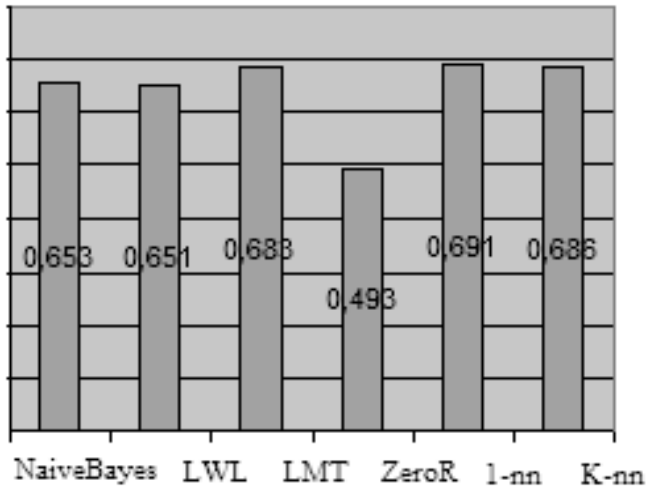


Gráfico 6. Comparación de efectividad entre métodos.

Los diez ejemplos no son suficientes para determinar qué tan efectivo o no es realmente uno u otro método, pero lo que sí nos permite es dar un vistazo inicial (nada definitivo) de la alta efectividad de clasificación de k-nn, de su comportamiento ante diferentes características de los datos y de sus resultados comparados con otros métodos. Por ejemplo, observemos en el gráfico 6 que k-nn obtuvo el mayor promedio de efectividad en la clasificación de los ejemplos seleccionados. Si comparamos las tablas 3 y 4 se nota que k-nn obtuvo su mejor rendimiento cuando el número de datos de entrenamiento era menor y que su peor desempeño fue en el ejemplo con mayor número de clases. También se puede ver cómo k-nn es el más efectivo cuando hay mayor número de atributos (el ejemplo kc) y cuando hay gran número de datos de entrenamiento (el ejemplo).

Como vemos, todos estos resultados parciales hacen que nos demos cuenta de la efectividad de k-nn, y lo interesante y útil que resulta el método para utilizarlo en algún proyecto de aplicación de inteligencia artificial.

8. Conclusiones

- k-nn es más efectivo cuando hay gran número de datos de entrenamiento.

- La métrica de distancia usada en el método de los k-vecinos es fundamental para obtener resultados deseados.

- Además de variar la métrica de distancia usada en el método k-nn, se puede cambiar la forma de determinar k bien sea con un diagrama de Voronoi, estableciendo un radio o seleccionando el primer vecino más cercano, sin que esto afecte la forma general del método.

- Cuando se selecciona el radio como método para determinar k, es conveniente realizar varias pruebas con el fin de determinar un valor heurístico adecuado para r con el propósito de obtener buenos resultados en la clasificación.

- La efectividad de k-nn depende en gran medida de la calidad de los datos de entrenamiento y de su coherencia.

- En cuanto a la eficiencia, el método será menos eficiente entre más datos de entrenamiento haya, pero será también más efectivo.

Bibliografía

Hernández Orallo, J.; Ramírez Quintano, M. J. y Ferri Ramírez, C. (2004). *Introducción a la minería de datos*. Madrid: Pearson Educación.

Rich, E. y Knight, K. (1994). *Inteligencia artificial*. McGraw-Hill.

Rolston, D. W. (1990). *Inteligencia artificial y sistemas expertos*. Bogotá: McGraw-Hill.

Rusell, S. y Norving P. (1995). *Inteligencia artificial. Un enfoque moderno*. Prentice Hall Hispanoamericana S.A.

Winston, P. H. (1994). *Inteligencia artificial*. Addison-Wesley Iberoamericana.